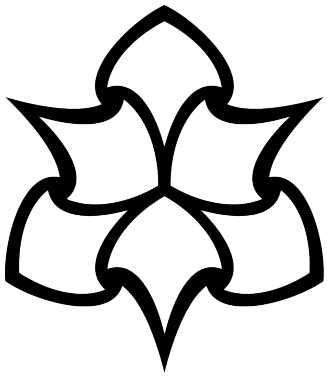


IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 5, 2020, accepted November 28, 2020. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.3042885

Semi-Supervised Hyperspectral Image Classification Based on Label Propagation via Selected Path

XILI WANG[1] AND HELEN JI[2]
[1]School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
[2]Department of Engineering, Manchester Metropolitan University, Manchester M15 6BH, U.K.

Corresponding author: Helen Ji (h.ji@mmu.ac.uk)

This work was supported by the National Natural Science Foundation of China under Grant 41471280, Grant 61701290, and Grant 61701289.

ABSTRACT Most graph-based semi-supervised classification methods do not perform well in hyperspectral image classification tasks due to their high complexity and other limitations. This paper proposes a label propagation semi-supervised classification algorithm which uses a few selected important paths to considerably reduce computational costs. It establishes that the most important paths for label propagation are minimum cost paths. It proves that minimum cost paths exist in minimum cost trees (MCT), and proposes a method based on a variant minimum spanning tree (MST) combined with priority queue to construct MCTs. The algorithm propagates labels from unlabeled nodes to labeled ones, a unique way different from any other studies where propagation is in the opposite direction, which brings about several clear advantages. These include that only one propagation path is required for each unlabeled node, improving both timing and memory performance. It also helps to solve a problem posed by sparse graphs where some image pixels cannot be classified, a situation which is especially problematic in large-scale image classification. The proposed method has the advantages of linear computational complexity, is independent of data dimension, has fewer parameters and is insensitive to the values of parameters. Moreover, it does not need large numbers of labelled pixels nor complex training processes. Experiments on hyperspectral images have shown that, compared with several existing algorithms, the proposed method achieves better performance in less time. The paper addresses some fundamental issues regarding propagating labels in graph based semi-supervised classifications. Due to the simplicity and the fast speed of the algorithm, it is also suitable to be integrated into both state-of-the-art and future hyperspectral image classification frameworks which have a label propagation stage.

INDEX TERMS Graph-based classification, hyperspectral image, label propagation, semi-supervised classification.

I. INTRODUCTION

Hyperspectral images (HSIs) have been widely used in earth observation, such as identification and classification of land covers. Supervised classification methods (e.g., Bayesian classifier, support vector machine, neural network) have been studied and applied extensively for such purposes. In order to achieve satisfactory accuracy and to improve generalization performance, large numbers of costly labeled samples are required in training. In addition, the high dimensional characteristic of HSIs brings computation difficulties for supervised classification methods. Thus, methods with properties of better classification ability and generalization performance, small labeled sample sets and low computational complexity independent of data dimension are desired.

Semi-supervised classification methods make use of both labeled and unlabeled samples in training. Because they use information derived from unlabeled samples, classification accuracy can be satisfactory even with a few labeled samples. When applied to remotely sensed images, these methods improved classification accuracy compared with supervised methods [1], [2]. However, most semi-supervised methods

The associate editor coordinating the review of this manuscript and approving it for publication was Long Xu.

have high spatial-temporal complexity since they use every labeled and unlabeled sample in training, thus limiting their application. Though using a portion of unlabeled samples is allowed in some methods, such as in LapSVM [3], selecting useful unlabeled samples becomes a new challenge.

Graph-based semi-supervised classification methods construct graphs from images in order to classify them. Label propagation is a typical semi-supervised classification method. It propagates labels from a small set of labeled data to a much larger set of unlabeled data in a graph. Graph construction and label inference are two main operations for this type of methods [4]. For the first operation, nearest neighbor (e.g. k-nearest neighbor, k-NN) methods are often used to construct graphs and the resultant graphs are often sparse. Sparse similarity matrixes derived from sparse graphs can reduce false connections between nodes, therefore help to obtain weights that represent actual similarity between nodes and prevent propagation of irrelevant information in graphs. Thus, sparse graphs are beneficial for decreasing computational costs and improving classification accuracy [5]. During the second operation, labels propagate among the nodes that belong to the same class according to a label inference method. Labels are able to spread to all the nodes in an undirected fully connected graph. Because a sparse graph may not be fully connected, especially for a large-scale image characterized by both a high degree of spatial aggregation and few neighbors, propagation processes may not be able to spread labels to all the nodes. This issue is seldom addressed by related research such as [6].

Label inference relies on an object function (also called energy function) defined on both labeled and unlabeled data. An energy function is established on two constraints. The first one restricts the misclassification cost of labeled data to the minimum. The second one is based on two assumptions of consistency - cluster assumption and manifold assumption. Excluding graph construction, the main differences between various semi-supervised classification methods lie in their different ways of defining and solving energy functions. Label inference methods can be divided into several categories based on the rules that govern how labels spread when solving a defined energy function. Though various methods appear different, they all share an identical framework, possess the property of label propagation, and some equivalence between them has been proved [7]. Another common property is high computational complexity. Assuming that k-NN is used to construct graphs, and that classification costs do not include the time of graph construction, for a problem with the category number c and the sample size n (include both labeled and unlabeled samples), the reported classification complexities of several well-known algorithms are as follows: Tikhonov Regularization and Manifold Regularization algorithm [8] and Spectral Graph Transducer algorithm [9] have $O(n^3)$; Gaussian Random Fields and Harmonic Functions [11], Local and Global Consistency [7] and LP (Label Propagation) [12] have $O(cn^2)$. It is clear that the high complexities limit their application in large-scale data classification.

Both decreasing classification complexity and extending semi-supervised classification methods for large-scale data applications have attracted much attention. Delalleu *et al.* [13] constructed a graph from a small subset of samples and proposed a nonparametric method to predict labels based on the sample subset. The training complexity of this method approximates to $O(m^2n)$ and the testing complexity is $O(mn)$, where m represents the size of the sample subset. Liu *et al.* proposed the Anchor Graph Regularization (AGR) algorithm [5]. AGR algorithm clusters data and defines the clusters' centers as anchors. It constructs Anchor Graphs to represent relationships between anchors and data. It then predicts an anchors' label according to Laplace manifold regularization algorithm. Finally, from an anchors' label and the Anchor Graph it can obtain labels for all data. The reduced Laplacian graph decreases the complexity of AGR to $O(m^2n)$, where m represents the number of anchors. Kim and Choi proposed Minimax Label Propagation (MMLP) [6] for scalable semi-supervised learning. MMLP estimates the distance of a path connecting two nodes in its label propagation process, blocks most of the propagation paths and only makes labels spread along a few important paths. In the best case, the time complexity of MMLP decreases to $O(n)$. It can be found that reducing the size of graphs or the number of propagation paths can effectively lower computational costs, otherwise, for large-scale datasets, computations may not even be feasible. But the above-mentioned literature were applied to small-scale datasets (hundreds or thousands samples). For large-scale datasets (more than hundreds of thousands of samples), m could still be a large value. Furthermore, the constructed sparse graphs bring about the connectivity problem (or rather the lack of connectivity problem, i.e. a graph is disconnected or weakly connected). This leads to the case that labels are not propagated to all nodes, leaving some data unclassified.

Propagating labels along just a few important paths between two nodes is an effective way to reduce computational costs. Cluster assumption indicates that adjacent nodes and nodes connected through high density areas are likely to have identical labels, which means that important propagation paths exist among these nodes. Inspired by these facts, this paper proposes a semi-supervised label propagation classification method via selected paths (SPLP). The contributions of this paper are as follows: (1) It establishes that the important paths for label propagation are the paths with minimal costs. (2) It proves that minimum cost paths exist in minimum cost trees (MCT), and proposes a method based on a variant minimum spanning tree (MST) combining with priority queue to construct MCTs. The MCTs are approximate but give linear performance with negligible loss of optimality. (3) It solves the connectivity problem brought about by sparse graphs, thus allowing all data to be classified. (4) It develops a new and unique way for label propagation. Unlike other label propagation methods, when searching for

a propagation path, SPLP starts from an unlabeled node and stops at a labeled node. This unique method of propagation brings about several clear advantages: a) it has superior spatial and temporal performance because only one path for each unlabeled node is computed and saved; in contrast, if labels propagate from labeled nodes to unlabeled ones, it is necessary to compare the cost of each of the paths from each labeled sample to the unlabeled sample, i.e. multiple paths need to be calculated and compared; b) it makes it possible to re-propagate unclassified samples,by picking one such sample and building the MCT for it using a larger K value. In other words, this unique way of propagation is the reason that (3) is achieved; c) empirical results shows that compared with other studies, it gives rise to fewer unclassified samples after the first iterative round.

The maximum classification complexity of SPLP is *O(nklogk)*, where k is the number of neighbors when constructing sparse graphs. Because k is very small comparing with n, SPLP has linear classification performance. Moreover, it is independent of data dimension and can classify multiclass directly, making it particularly suitable for HSI classification that is characterized by high dimension, multiclass, large-scale, and small numbers of labeled samples.

The rest of this article is organized as follows: Section II provides the details of the proposed method and the solutions to the key problems. Section III described the experiments conducted by using SPLP on three well-known HSIs data sets, and compares and analyses the results against three competing algorithms. This section also compares SPLP with some state of the art research in HSI classification. Finally in Section IV, conclusions are drawn based on the comparisons.

II. METHOD

A. LABEL PROPAGATION

In this section, we explain what kind of path will be selected for label propagation and how to find such a path. Specifically, unlike other methods, path searching starts from unlabeled samples rather than labeled ones.

Take a sample set $\boldsymbol{X} = \{x_i\}_{i=1}^{n} \subset \boldsymbol{R}^d$, whose first l ($l \ll n$) samples are labeled, and the rest are unlabeled. If each sample belongs to one of the c classes, the aim of semi-supervised classification is to find the labels for the unlabeled samples $\{\boldsymbol{x}_i\}_{i=l+1}^{n}$.

Graph based semi-supervised classification methods first construct an undirected graph $\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{E})$ for dataset $\boldsymbol{X}$. k-NN is usually used for this construction purpose. In the graph, nodes x_i (i.e. spectral information of pixel i) represent data in the dataset. These nodes are linked by edges $\boldsymbol{E} = \{(x_i, x_j)\}$ only if they are k-NN of each other in the dataset. When K is small (5-20) sparse graphs are constructed. Sparse similarity matrixes generated from sparse graphs help to obtain more accurate weights than from dense graphs due to fewer false connections between nodes. Using sparse graphs not only reduces the amount of computation, but also helps to improve classification accuracy [5]. Weight w on an edge measures similarity between the connected nodes. It is obtained by Gaussian Kernel Function:

$$w_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{\sigma^2}) \tag{1}$$

Here bandwidth parameter $\sigma > 0$.

Define a c-dimensional real function $f_i \in \boldsymbol{R}^c$ as representing "soft assignments" of each node x_i to C classes. Define a c-dimensional binary vector $y_i \in \{0, 1\}^c$ as a "hard assignment" of a label for node x_i to its true class label. If x_i belongs to the pth class ($p \in (1, \cdots, c)$), then pth dimension of y_i is set to 1, otherwise set to 0. Following [6], define an objective function $E(f)$ as in (2). Every unlabeled sample gets its optimal label when $E(f)$ reaches its minimum:

$$E(f) = \sum_{(i,j)\in E} w_{ij} \|f_i - f_j\|^2 \quad s.t. f_i = y_i \, for \; i = 1, \ldots, l \tag{2}$$

Minimizing (2) is computationally expensive (in the order of $O(cn^2)$). The next section proposes a selective path based approach to predict labels.

B. LABEL PROPAGATION VIA SELECTED PATH

Define a set of paths existing between any two nodes x_i and x_j as:

$$A_{ij} = \{a = (a_0, a_1, \ldots, a_m) \, | m \geq 1, \forall \ell \in [0, m-1], \\ (a_\ell, a_{\ell+1}) \in \boldsymbol{E}, a_o = i, a_m = j, a_1, \ldots, a_{m-1} \neq j\} \tag{3}$$

Path $a \in A_{ij}$ connects nodes x_i and x_j via several consecutive edges. If the cost of an edge on the path is $c(a_l, a_{l+1}) = w_{l,l+1}$, define the overall cost of the path as the P-norm of the edge costs alone the path:

$$\|c(a)\|_p = \left[\sum_l c(a_l, a_{l+1})^p\right]^{1/p} \tag{4}$$

Define the sum of costs of all the paths from x_i to all the labeled data as:

$$c(x_i) = \sum_{j=1}^{l} \sum_{a\in A_{ij}} \exp\left(-\tfrac{1}{T} \|c(a)\|_p\right)$$

Then to compute f_i for unlabeled nodes i, we have

$$f_i = \sum_{j=1}^{l} \sum_{a\in A_{ij}} \exp\left(-\tfrac{1}{T} \|c(a)\|_p\right) y_j$$

Here P and T are two important parameters. When $P \to \infty$, the costs of the paths via lower density regions tend to be larger and via higher density regions tend to be smaller [6]. The decay constant, $0 < T < \infty$, represents a preference for paths of smaller total costs. According to the cluster assumption, when $T \to 0$, costs of a small number of paths can represent the total costs of the paths to all the labeled samples. Furthermore, according to [6] when $T \to 0$ and $P \to \infty$, $c(x_i) \to \min_{j=1,\ldots,l, a\in A_{ij}} \|c(a)\|_p$, i.e. the total costs of paths starting from x_i to all the labeled samples approximates

to the minimum path cost among all such path costs. Thus we have:

$$f_i \to y_{j^*}, j^* = \arg \min_{j=1,..,l, a \in A_{ij}} \|c(a)\|_p \tag{5}$$

Formula (5) shows that the label for x_i should be the same as the label for x_j, where x_j is the labeled sample which makes $\|c(a)\|_p$ minimum. Then the goal of the algorithm is to find the minimum cost path from the unlabeled sample x_i to all the labeled samples.

When $P \to \infty$, for any path $a \in A_{ij}$, according to the definition of path cost (4), there is:

$$\|c(a)\|_p \to \max c(a_l, a_{l+1}) \tag{6}$$

Equation (6) means that the cost of a path is approximately equal to the maximum edge cost on this path. Thus for the efficiency of computation, we use the largest edge cost on a path to represent the cost of the whole path.

C. FINDING SELECTED PATHS

This section addresses how to find the approximate minimum cost path defined by (6). It proposes a method of using a variant minimum spanning tree (MST) combined with priority queue to construct MCTs.

According to the principle of minimum spanning trees: in a graph $\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{E})$, $\boldsymbol{U}$ is a nonempty subset of $\boldsymbol{X}$ and represents the set of nodes in its MST. There is a node $u \in \boldsymbol{U}$, and $v \in \boldsymbol{X} - \boldsymbol{U}$ is a node outside $\boldsymbol{U}$. If (u, v) is an edge in $\boldsymbol{G}$, and the cost of the edge (u, v) is the minimum cost among the costs of all the edges directly connected to $\boldsymbol{U}$, then the next node to be added to the MST would be $\boldsymbol{v}$. In other words, the MST generated from graph $\boldsymbol{G}$ is a set of edges that have the minimum sum of costs when connecting all the nodes.

Prim algorithm can be used to construct a MST, but it has a relatively high computational cost. Therefore, this paper proposes a method of combining priority queue with Prim algorithm to construct MCTs. By pre-ordering all the nodes to be added to a tree and placing them in a priority queue, it speeds up the search procedure, thus reducing time complexity.

The MCT construction procedure by priority queue and Prim algorithm is as follows:

(1) Starting from any unlabeled sample x_i, add x_i to the MCT set U, add the edges between x_i and its k nearest neighbors x_j ($x_j \in \boldsymbol{X} - \boldsymbol{U}$) to a priority queue in the order of their similarity weights w_{ij}.
(2) Pop out the head of the queue and examine its node x_e. If $x_e \notin \boldsymbol{U}$ then add it to $\boldsymbol{U}$. If x_e is an unlabeled sample, add its edges to the priority queue in the order of their similarity weights. Repeat (1) and (2). If x_e is a labeled sample, goto (3).
(3) This MCT is constructed. Assign the label of all the unlabeled samples in the tree with the label of x_e.
(4) If there are any unlabeled samples left, pick one randomly and repeat the steps (1)-(3) to construct another tree until every unlabeled sample is in a MCT.

Obviously, this algorithm constructs multiple trees with each of them containing a fraction of the nodes, instead of a single tree containing all the nodes.

Note that this algorithm actually generates approximate MCTs. The length of a queue is usually much smaller than the total number of edges in a graph. Therefore with the progress of the algorithm, and more and more edges being added in, it is likely that the edges at the back of the queue could be gradually discarded due to the limited queue length, meaning potential minimum cost paths could also be discarded. However, the impact of the loss of optimality is regarded as negligible by this study because 1) each node has maximum of k (the number of nearest neighbors) edges. k is normally a very small number compared with the number of edges in a graph. If a queue is long enough, the queue might not even be filled up when the algorithm terminates, meaning there is no loss of optimality. Even if the queue is filled up and some edges are discarded, these are the edges with the lowest priority at the back of a queue, so the impact would be small. 2) this algorithm does not aim to generate a full MCT for a given graph, rather it terminates when a labeled node is added into the tree. As the impact of limited queue length mostly takes effect towards the later stage of computation, it is likely that such an effect would not have materialized before the computation terminates.

The rest of this section proves that within a MCT, the only path from an unlabeled node to a labeled node is the minimum cost path between the two nodes.

Construct a MCT T starting from any unlabeled node x_i in graph $\boldsymbol{G}$ till a labeled node x_j is added. Assume that the minimum cost path from x_i to x_j is not in T, then there must exist a path $a \in A_{ij}$, such that $\|c(a)\|_p < \left\|c(T_{ij})\right\|_p$, where $\left\|c(T_{ij})\right\|_p$ is the cost of path from x_j to x_i in T, and $\|c(a)\|_p = \max_l c(a_l, a_{l+1})$, $\left\|c(T_{ij})\right\|_p = \max_\tau c(a_\tau, a_{\tau+1})$. Here $c(a_\tau, a_{\tau+1})$ is the cost of connecting adjacent nodes in the path from x_i to x_j in T after x_j joins T.

The proof is by contradiction under two different cases: (1) Assume that the whole minimum cost path is not in the MCT. (2) Assume that part of the minimum cost path is in the MCT.

Under the first case: assume in $\|c(a)\|_p$, sample x_k connects x_i, and x_k is not in the MCT. According to the rules of MST, there is $c(a_i, a_k) \geq \max c(a_\tau, a_{\tau+1}) = \left\|c(T_{ij})\right\|_p$. Because $c(a_i, a_k) \leq \|c(a)\|_p$, therefore $\|c(a)\|_p \geq \left\|c(T_{ij})\right\|_p$. This contradicts the assumption.

Under the second case: if $\|c(a)\|_p$ connects to a node x_{k+1} in T via a node x_k $\left(x_k \neq x_j\right)$ which is not in T, then according to the rules of MST there is $c(a_k, a_{k+1}) \geq \max c(a_\tau, a_{\tau+1}) = \left\|c(T_{ij})\right\|_p$. Because $c(a_k, a_{k+1}) \leq \max_\ell c(a_\ell, a_{\ell+1}) = \|c(a)\|_p$, therefore $\|c(a)\|_p \geq \left\|c(T_{ij})\right\|_p$. This contradicts the assumption as well.

Therefore, the minimum cost path between an unlabeled node and a labeled node is in the MCT containing these two nodes.

D. THE CONNECTIVITY PROBLEM OF SPARSE GRAPHS

As previously mentioned, sparse graph cannot guarantee full connectivity. When the number of nearest neighbors is small, or the data scale is large and data cluster together (e.g. in large-scale image), it is common to form several unconnected groups within a sparse similarity matrix, which blocks label propagation and results in some samples not being classified. This problem has barely been analyzed and solved so far.

There are two kinds of connectivity problems. One is brought by disconnected sparse graph, as shown in Fig. 1.

In Fig. 1, the red and green nodes represent labeled samples in class 1 and class 2 respectively, and black nodes are unlabeled samples. Each node in this graph connects to only two nearest neighbors, resulting in one sparse similarity matrix becoming three disconnected clusters. The three black nodes in the middle cluster will not be classified by label propagation since they are not connected to any labeled samples.

This paper proposes a method of re-propagation to solve this problem: after one round of propagation, it uses KD tree nearest neighbor searching algorithm [14] to extend the searching area to reach the nearest neighbors for those unclassified nodes. In this round the number of neighbors is twice the number in the previous round. Although the number of neighbors is increased, the algorithm only searches for the neighbors of some specific data within the whole data set, hence it does not increase computational complexity much. If there are still unreached nodes, repeat the above procedure until all the nodes are reached. Note that this approach of re-propagating labels by using larger K values is only possible because SPLP propagates from unlabeled nodes.

Another connectivity problem is brought by weak connectivity in sparse graphs: sparse matrixes constructed from k-NN could be asymmetric. This means that node x_i is one of the k nearest neighbors of node x_j, but x_j is not one of the k nearest neighbors of x_i. In the corresponding sparse graph there exists a path from x_j to x_i, but there is no path from x_i to x_j. Such a graph is called a weakly connected graph. Existing algorithms cannot classify those weakly connected nodes.

Though increasing the number of nearest neighbors in sparse graphs may solve the weak connectivity problem, the authors found that small increments cannot guarantee to solve the problem, but will increase the time complexity of the overall classification algorithm. Another solution is to change connections to be symmetric. This solution redefines the weights, making the similarity matrix symmetrical:

$$w_{ij} = \begin{cases} w_{ij}, & if\ w_{ij} = w_{ji} \\ w_{ij}, & if\ w_{ij} > 0\ and\ w_{ji} = 0 \\ w_{ji}, & if\ w_{ij} = 0\ and\ w_{ji} > 0 \end{cases} \tag{7}$$

After this weight redefinition, labels can propagate in any direction, and all nodes can be reached in weakly connected sparse graphs.

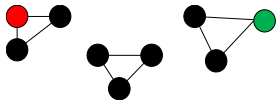


FIGURE 1. Illustration of the problem caused by disconnected graphs.

E. COMPUTATIONAL COMPLEXITY ANALYSIS OF SPLP

The algorithm mainly has three parts: 1) constructing a graph and calculating a similarity matrix; 2) constructing MCTs to propagate labels; 3) re-propagating labels for those unlabeled samples from the previous propagations.

The first part uses FLANN algorithm [14] to construct sparse nearest neighbor graphs. Its time complexity is $O(ldnlogn)$, where l is the number of trees, and d is the number of data dimensions.

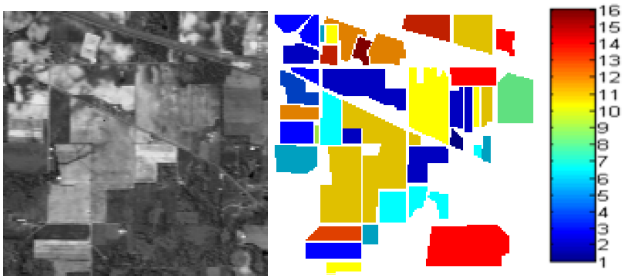
In the process of constructing MCTs, the algorithm uses priority queues to sort data. The complexity for initially building a priority queue of length n_q *is* $O(n_q)$. The complexity of popping one sample from a priority queue is $O(1)$. The maximum complexity of re-sorting the queue after adding another sample's edges is $O(n_q \log n_q)$. The maximum complexity of adding one node to the MCT by marking it as in the MCT set is $O(1)$. Therefore the maximum time complexity of constructing a MCT is $O(n_t n_q \log n_q)$, where n_t is the number of nodes in a MCT. The maximum time complexity of constructing all MCTs is $O(\sum_{i=1}^{p} n_{t(i)} n_{q(i)} \log n_{q(i)})$, where p is the number of MCT trees. When the same queue length is used for all trees, it becomes $O(nn_q \log n_q)$. For graphs constructed by kNN, it is convenient to have n_q take the value of k, i.e. the length of priority queues is taken as the number of the nearest neighbors, the complexity is $O(nk \log k)$.

In the third part, assume the number of data that need to be re-propagated is m. Because FLANN algorithm only needs to calculate KD trees when first constructing a graph, thus the graph constructed in part 1) is reused for this step. The time complexity of constructing new MCTs is $O(mk_1 \log k_1)$ (k_1 is the number of the nearest neighbors for re-propagating). Because n is much greater than m and k_1, the complexity of the third part has $O(mk_1 \log k_1) < O(nk \log k)$.

In summary, the time complexities of the three parts of SPLP are $O(ldn \log n)$, $O(nk \log k)$ and $O(mk_1 \log k_1)$ respectively. Thus, if ignoring the graph construction part which is common to most published algorithms, the complexity of the classification part of SPLP is $O(nk \log k)$. As k is usually a constant and is far smaller than n, it can be said that the time complexity of SPLP is linear.

III. EXPERIMENTAL RESULTS AND ANALYSIS

This section shows the performance of the semi-supervised SPLP classifier on three common and challenging hyperspectral image datasets, and compares it with three competing methods: LP, AGR and MMLP. The algorithm is implemented by MATLAB and C++ on a computer equipped with an Intel Xeon E5504 Processor (2.0 GHz) and 8G memory.



(a) (b)

FIGURE 2. Indian Pines. (a) Original image. (b) Ground truth.

A. HYPERSPECTRAL DATA

The first dataset was obtained by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines region in Northwestern Indiana in 1992. The hyperspectral imagery has a spatial size of 144 * 144 pixels with 20m spatial resolution. 24 spectral bands among the total 224 spectral bands were removed due to noise and water absorption phenomena. The rest of the bands were used in classification. There are 16 mutually exclusive classes. The image and the reference land cover are shown in Fig. 2.

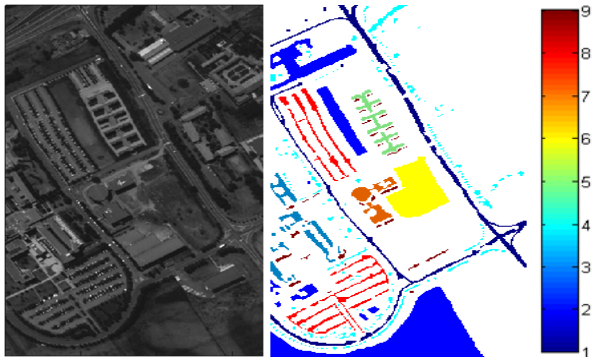
The second dataset was obtained by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over the University of Pavia. 103 bands were used after water absorption bands were removed from the original 115 bands. The size and the spatial resolution of the image are 610*340 and 1.3m respectively. Fig. 3 shows the image and the 9 reference land cover classes.

The third dataset was collected by AVIRIS over Salinas Valley, Southern California, in 1998. The area contains a spatial size of 512 * 217 pixels and spatial resolution of 3.7m. 204 spectral bands were left after discarding 20 water absorption bands. There are 16 classes of the land covers. The image and the reference land cover map are shown in Fig. 4. The number of the labeled samples of the images and other details can be obtained from literature [15].

B. CLASSIFICATION RESULTS AND ANALYSIS

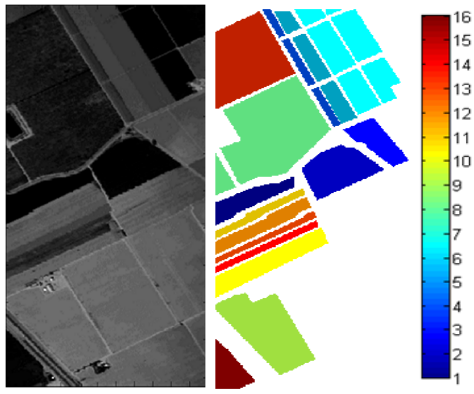
The results of SPLP are compared to that of LP, AGR and MMLP. In the experiment, according to (1), the weight of an edge is defined as: $w(i,j) = \exp(-\beta\|x_i - x_j\|^2)$, in which $\beta = 2\beta_o^{-1}$. β_o is the mean value of the squares of the distances between each sample and its 10th neighbor. k in k-NN is assigned to 20. The influence of k is tested and analyzed in the experiment. The parameter S (represents the number of neighbor anchors of each sample) in AGR is set to 3. The number of anchors in AGR is set to 40 times of the number of categories.

For each hyperspectral image, the experiment randomly selects 5%, 10%, 15% and 20% of the labeled samples from each category as training samples. The rest are unlabeled samples. Training samples are the same for all the methods. The classification results evaluated by overall accuracy (OA) and Kappa coefficient (KC) are shown in Tables 1, 2 and 3.



(a) (b)

FIGURE 3. Pavia University. (a) Original image. (b) Ground truth.



(a) (b)

FIGURE 4. Salinas. (a) Original image. (b) Ground truth.

TABLE 1. Classification results of OA and KC of Indian Pines.

		Labeled samples per class			
		5%	**10%**	**15%**	**20%**
LP	OA(%)	71.23	74.67	76.11	79.35
	KC	**0.68**	**0.73**	**0.76**	0.79
AGR	OA(%)	56.32	59.42	62.56	64.05
	KC	0.54	0.59	0.64	0.68
MMLP	OA(%)	69.16	72.33	72.93	76.40
	KC	0.66	0.71	0.74	0.79
SPLP	OA(%)	**71.35**	**75.37**	**77.61**	**81.58**
	KC	0.67	**0.73**	0.75	**0.81**

The classification time of each method (using 20% of the labeled samples) is listed in table 4. The differences in running time caused by the different numbers of labeled samples are less than 0.1 second for SPLP and MMLP, and less than 0.5 second for AGR and LP. Fig. 5 to 7 show the classification results.

TABLE 2. Classification results of OA and KC of Pavia University.

		Labeled samples per class			
		5%	10%	15%	20%
LP	OA(%)	83.51	85.64	87.64	88.29
	KC	0.79	0.80	0.83	0.85
AGR	OA(%)	63.91	66.66	68.99	69.15
	KC	0.58	0.63	0.67	0.69
MMLP	OA(%)	82.26	83.72	84.65	85.45
	KC	0.78	0.81	0.83	0.85
SPLP	OA(%)	**85.08**	**86.54**	**87.67**	**88.78**
	KC	**0.80**	**0.83**	**0.86**	**0.87**

TABLE 3. Classification results of OA and KC coefficient of Salinas.

		Labeled samples per class			
		5%	10%	15%	20%
LP	OA(%)	88.81	89.60	90.24	91.23
	KC	0.87	0.88	**0.90**	0.91
AGR	OA(%)	84.79	85.18	85.36	85.91
	KC	0.84	0.85	0.86	0.87
MMLP	OA(%)	87.56	88.04	88.79	89.78
	KC	0.87	0.88	**0.90**	0.90
SPLP	OA(%)	**89.28**	**89.93**	**91.25**	**92.16**
	KC	**0.88**	**0.89**	**0.90**	**0.92**

TABLE 4. Classification time (second) of the three images (using 20% of the labeled samples).

	LP	AGR	MMLP	SPLP
Indian Pines	3.40	9.98	**1.23**	1.25
Pavia University	63.21	46.23	**4.26**	4.33
Salinas	109.43	39.87	**6.72**	7.40

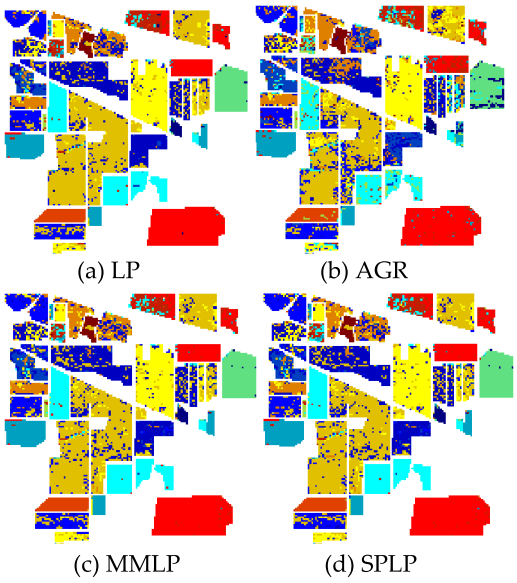


FIGURE 5. Classification results of Indian Pines. (using 20% of the labeled samples).

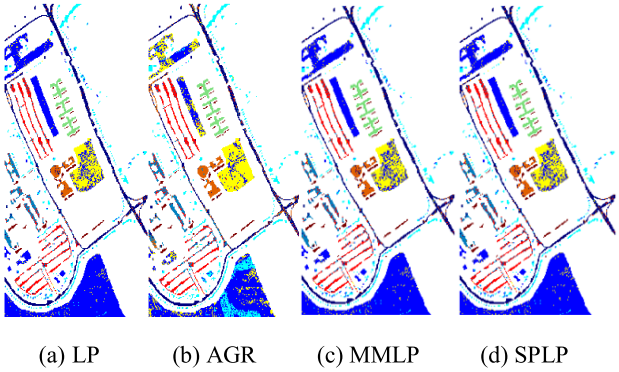


FIGURE 6. Classification results of Pavia University. (using 20% of the labeled samples).

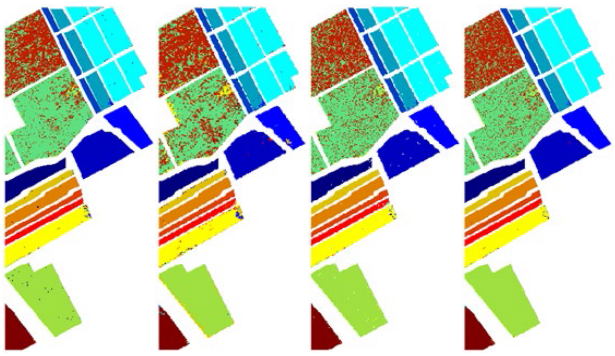


FIGURE 7. Classification results of Salinas. (using 20% of the labeled samples).

From the results we can see that SPLP has the highest values for both OA and KC, which indicates that SPLP has the best classification performance. LP also does well since labels are spread through every possible path. The results of MMLP are inferior to that of SPLP and LP. Taking Indian Pines as an example, there are 310, 299, 276 and 207 pixels that are not classified in MMLP when 5%, 10%, 15% and 20% labeled samples are used. In particular, the KC values of MMLP turn to be superficially higher since the data that are not classified decrease the denominator in

TABLE 5. Impacts on the results by different k in k-NN (Pavia University, using 20% of the labeled samples) ① No. of unclassified data. ② No. of unclassified data after the first propagation round.

K		20	40	60	80
	①	199	31	15	7
MMLP	OA (%)	85.45	85.77	86.17	86.66
	Time (s)	4.26	4.75	5.32	5.68
	②	**104**	**11**	**0**	**0**
SPLP	OA (%)	88.78	88.78	88.79	88.79
	Time (s)	4.33	4.88	5.46	5.89

KC expression. In terms of accuracy, the worst performer is AGR. The classification results of AGR are influenced by an important parameter—the number of anchors, which is often not the best when chosen manually.

The data to be classified are the unlabeled data in each image. The ascending order of data size is: Indian Pines, Pavia University and Salinas. From the results, we can see that the classification time increases as the size of the datasets increase. The computational complexities of MMLP and SPLP are approximately linear to data size, which confirms the analysis in section *II.E*. SPLP has slightly longer run times than MMLP does because it goes through re-propagation iterations in order to classify all pixels. In AGR, the number of anchors increases with the increase of the number of categories and the size of images, thus AGR costs much more time than MMLP and SPLP. For LP, the classification time increases rapidly as the data size increases, showing that it has the highest complexity due to labels propagating along every possible path.

k-NN is used to construct graphs. k value will affect the connectivity of sparse graphs. A small k value will help to reduce classification time but may increase the number of data not being classified. Large k values may solve the problem, but will increase classification time. Table 5 lists the classification results and time, and the number of data that are not classified for different k values in Pavia University. It can be seen that SPLP solves the problems encountered, thus a small k value is enough. It also shows that when labels propagate from unlabeled nodes to labeled ones, the number of unclassified data is significantly reduced after the first propagation round. Furthermore, the re-propagation step makes the algorithm almost insensitive to the value of parameter k, which is a good property as it allows to conveniently choose a small k value. In contrast, MMLP still has data that are not classified even when a large k value is set.

The proposed SPLP chooses important paths for label propagation to reduce time complexity. The time complexity of SPLP is approximately linear to the size of data. It is substantially faster than LP and AGR. It is also more robust since it is not affected by algorithmic parameters used by LP and AGR. Compared with MMLP algorithm, SPLP is more accurate and clarifies that propagation paths are the minimum cost paths, so that each unlabeled node only needs one propagation path and each node is labeled only once. In contrast, MMLP relies on propagation via multiple paths. Although it quickly cuts off those unimportant propagation paths to reduce the amount of propagation, it still has higher memory costs because multiple paths need to be saved for each unlabeled node. In summary, SPLP algorithm is superior in terms of defining and searching for important paths. It has better memory performance. In addition, SPLP solves the problem of some data not being classified due to weakly connected or disconnected graphs, thus further improving classification performance.

C. WIDER COMPARISONS

In Section *III.B*, the experimental results of SPLP are compared with three directly related algorithms, namely LP, AGR and MMLP. This section compares SPLP with some important state-of-the-art (SOTA) works in the wider area of classification of hyperspectral images. Almost all of the articles cited in this section used at least one of the datasets used by this research. Due to the differences in experiment environment and settings, and the difference in how the parameters were set up, results from these literatures are not directly comparable with each other and with SPLP. Whenever appropriate, OA figures are quoted to indicate classification accuracies and to provide an overall picture of the current state of research.

One of the current state-of-the-art methods of HSI classification focuses on extracting more effective features using various deep learning networks. In [16] (MHRNN as the acronym for the algorithm) 3D CNNs are applied on multi-scale local image patches to extract the multi-scale local spectral-spatial features. This is followed by constructing multi-scale 1D sequences in eight directions on the 3D local feature domain, then multi-scale hierarchical recurrent neural networks (MHRNNs) are used to learn the spectral-spatial features at different domains. The research reported high OA and AA accuracies for Pavia University and Salinas data sets, in the range of over 98% using 1% of training samples. The authors commented that their method takes more time to run compared with other deep learning based HSI classification methods, but no run time was presented in the paper. In [17] (CAG) and [18] (SAGP), improvements to the traditional attention mechanisms were made by using cross-attention mechanism and graph convolution integrating algorithms. [18] further adopted a bidirectional independent recurrent neural network to extract features. Reference [17] reported OA accuracy for Indian Pines data set as 77% using 5% of training samples. In contrast, SPLP achieves 71.35% OA accuracy for India Pines using 5% of labeled samples. The run time in [17] for the above mentioned experiment is 1.16s on a computer equipped with an

additional NVIDA GTX10170 GPU to accelerate the computation. In contrast, SPLT's run time is around1.25s on a modest personal computer without GPU. Reference [18] uses completely different datasets so the results are not comparable. From the results, one can see that although SPLP only uses original spectral features and does not rely on complicated training processes, its accuracy is comparable with some of the state-of-the-art deep learning based approaches. In terms of run time, SPLP is far more superior, because deep learning based approaches need to take advantages of accelerations hardware (such as GPUs) to be on par with SPLP.

Another state-of-the-art method used by many researchers to reduce complexity centers around the idea of superpixel or subspaces. It is typically used in conjunction with other multiple steps to improve classification accuracy. In [19] (MSSR), the authors used multi-scale superpixels to explore the spatial information of HSIs. It then uses joint sparse representation classification (JSRC) to classify the multi-scale superpixels. This is followed by a majority voting to fuse the labels of superpixels at different scale and to obtain the final classification result. The best OA classification accuracy achieved by [19] for India Pines dataset when 10% of labels were sued as training set is 98%. This is much higher than SPLP's 75.37% when 10% of samples are labeled. The experiments in [19] are conducted on an Inter® Xeon® CPU E5-2603 V3 @1.60 GHZ and 96 GB of RAM. For the India Pines result mentioned, it took 67.4s, much higher than SPLP's run time (in the region of 1.25s). This is despite the fact that the computer used in [19] has a much higher performance CPU (according to various CPU benchmark websites such www.cpubenchmark.net) and a much larger memory (96GB vs 8GB). In [20] (SuWLP), entropy rate segmentation (ERS) was used to oversegment an image to form superpixels. It then designed a new similarity measure to estimate the similarity between two superpixels. Training samples also have to be expended based on superpixel distributions. A weighted label propagation was used to propagate labels at the superpixel level. Finally the labels on superpixels are mapped back to the original pixels. The experiments in [20] focused on a very low number of label samples: only 3 labeled samples per class. It achieved OA accuracies of 69.17%, 70.29% and 89.09% for India Pines, Pavia University and Salinas respectively. There was no run time reported in this paper. In [21], the random subspace method is used to partition the feature space into multiple subspaces, then multiple label propagation models are constructed on subspaces. Pseudo-labels were obtained after fusion decisions of the multiple labels. Extreme learning machines are trained on both labels and pseudo-labels. For their experiments, different numbers of labeled samples, ranging from 3 per class, to 200 per class, were chosen. For 3 labeled samples per class, [21] achieved OA accuracies of 79.62%, 85.54%, and 89.54% for India Pines, Pavia University and Salinas respectively; for 25 labeled samples for class, the figures are 92.9%, 95.93% and 96.44%. The results for other numbers of labeled samples are incomplete, but seem to have the trend of diminishing returns when the sample sizes increase. No run time is reported in the paper.

In summary, the current trend of semi-supervised HSI classification incorporates multiple steps and techniques to achieve high classification accuracy and use very small numbers of labeled samples. Many also resort to complicated parameter analysis and trials to achieve the best results. The reported run times are in general very high, and some need to employ hardware accelerations to be viable in terms of run time.

IV. CONCLUSION

Although SOTA is discussed in the previous section to provide an overall picture of the current state of the research, competing for higher classification accuracies and smaller sample sizes is not the research aim of this article. It becomes apparent from the comparisons in section *II.C* that a simple and straightforward method such as SPLP is unlikely to beat current SOTA. Instead this research intents to address some fundamental questions regarding propagating labels in graph based semi-supervised classifications, i.e. which are the important paths for propagation? how to propagate the labels (e.g. algorithms and direction of propagation)? and how to solve the problems that some data can never be classified because they are not reachable in the previous studies. All the questions are answered in this paper. All the experiments conducted here only use the original spectral information, and still it achieved high accuracy. SPLP can classify multiclass directly and is independent of data dimensions. Rigorous run time complexity analysis shows that it is linear to data size, therefore is extremely fast. It also has low spatial complexity not only because each unlabeled node needs only one propagation path, but also because these paths can be quickly cut off when a label is found. SPLP almost does not rely on parameters, making it very easy to use and be integrated with other techniques. Last but not the least, it is capable of reaching all the nodes to achieve a complete classification.

The above mentioned properties mean that SPLP can be readily integrated into state-of-the-art and future HSI classification frameworks which has a label propagation stage. In fact, the research team is working on integrating feature extractions and adjacent matrix graph learning with SPLP to further improve the performance of HSI classification.

ACKNOWLEDGMENT

The authors want to express their appreciation of Mr. Hongshuai Lin's assistance in coding.

REFERENCES

[1] X. Wang, Y. Hu, and S. Zhang, "A novel graph based label propagation method for hyperspectral remote sensing data classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Valencia, Spain, Jul. 2018, pp. 2579–2582.

[2] B. Du, T. Xinyao, Z. Wang, L. Zhang, and D. Tao, "Robust graph-based semisupervised learning for noisy labeled data via maximum correntropy criterion," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1440–1453, Apr. 2019.

[3] L. Gomez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe, "Semisupervised image classification with Laplacian support vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 336–340, Jul. 2008.

[4] O. Zoidi, E. Fotiadou, N. Nikolaidis, and I. Pitas, "Graph-based label propagation in digital media: A review," *ACM Comput. Surv.*, vol. 47, no. 3, 2015, Art. no. 48.
[5] W. Liu, J. He, and S. F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 679–686.
[6] K.-H. Kim and S. Choi, "Label propagation through minimax paths for scalable semi-supervised learning," *Pattern Recognit. Lett.*, vol. 45, no. 1, pp. 17–25, Aug. 2014.
[7] D. Zhou, O. Bousquet, T. N. Lal, and J. Weston, "Learning with local and global consistency," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2003, pp. 321–328.
[8] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.
[9] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proc. 20th Int. Conf. Int. Conf. Mach. Learn.*, 2003, pp. 290–297.
[10] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 19–26.
[11] X. Zhu and J. Lafferty, "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 1052–1059.
[12] X. Zhu, "Learning from labeled and unlabeled data with label propagation," in *Proc. Int. Joint Conf. Neural Netw.*, 2002, pp. 2803–2808.
[13] O. Delalleau, Y. Bengio, and N. L. Roux, "Efficient nonparametric function induction in semi-supervised learning," in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, 2005, pp. 96–103.
[14] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.
[15] *GIC*. Accessed: Dec. 8, 2020. [Online]. Available: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Pavia_Centre_and_University
[16] C. Shi and C.-M. Pun, "Multi-scale hierarchical recurrent neural networks for hyperspectral image classification," *Neurocomputing*, vol. 294, pp. 82–93, Jun. 2018.
[17] W. Cai and Z. Wei, "Remote sensing image classification based on a cross-attention mechanism and graph convolution," *IEEE Geosci. Remote Sens. Lett.*, early access, Oct. 1, 2020, doi: 10.1109/LGRS.2020.3026587.
[18] H. You, S. Tian, L. Yu, and Y. Lv, "Pixel-level remote sensing image recognition based on bidirectional word vectors," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1281–1293, Feb. 2020.
[19] S. Zhang, S. Li, W. Fu, and L. Fang, "Multiscale superpixel-based sparse representation for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 2, p. 139, Feb. 2017.
[20] S. Jia, X. Deng, M. Xu, J. Zhou, and X. Jia, "Superpixel-level weighted label propagation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 5077–5091, Jul. 2020.
[21] Y. Zhang, G. Cao, A. Shafique, and P. Fu, "Label propagation ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3623–3636, Sep. 2019.

XILI WANG received the B.S. degree in computer application technology from Tianjin University, Tianjin, China, in 1991, and the M.S. degree in computer application technology and the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 1994 and 2004, respectively.

Since 1994, she has been a Teaching Staff with Shaanxi Normal University, where she became a Professor with the School of Computer Science, in 2006. From 2011 to 2012, she had been a Visiting Scholar with the Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY, USA. She is the author of more than 90 articles and more than ten patents. Her research interests include artificial intelligence, machine learning, intelligent perception and understanding, image processing and analysis, and their remote sensing image application.

HELEN JI received the B.Eng. degree in computer science and engineering from Xi'an Jiaotong University, Xi'an, China, in 1991, the M.Sc. degree in computer science from Xidian University, Xi'an, China in 1994, and the Ph.D. degree in formal methods in hardware design from Stirling University, U.K., in 2000.

Since 2007, she has been a Senior Lecturer with the Department of Engineering, Manchester Metropolitan University, Manchester, U.K. She worked for Cadence Design Systems, Scotland, as a Member of Technical Staff before she moved to academia. Her research interests include formal methods, electronic design automation, machine learning, image processing, and embedded systems design.